Table of Contents

Interoperability and Compatibility in cloud APIs

This section is focused in the EC2 API, as it is the de facto standard for cloud API.

FermiCloud (OpenNebula) ECONE EC2 interface

It is possible to manage ONE using an EC2 interface (tested on ONE4.2). As a client one can use the euca2ools[1].
The definition of EC2 templates (and the contextualization) are very similar to ONE.

Here are the minimalistic instructions to get the EC2 interface working for OpenNebula4.2:

- install the EC2 interface on ONE4.2 host with

  ```
  # /usr/share/one/install_gems
  ```

- edit the file */etc/one/econe.conf* and choose the host that will listen the EC2 requests. For example:

  ```
  …

  # Host and port where econe server will run
  :host: fermicloudpp003.fnal.gov
  :port: 4567

  …
  ```

- start (or restart) the econe server (as oneadmin):

  ```
  $ econe-server start
  ```

- now you can access the EC2 interface from another host with euca2ools. In your PC, set up the environment:

  ```
  $ export EC2_URL=http://fermicloudpp003.fnal.gov:4567   # ----> find this in the :host: and :port: parameters. Note that OpenNebula is not completely following the standard here, so one must use the same host string as used in the server (i.e.: one can not interchange hostname/ip/alias)

  $ export EC2_ACCESS_KEY=oneadmin   # ----> username in ONE, find this with $oneuser list

  $ export EC2_SECRET_KEY=87683717368fda7c0dac905ab576d8a72836be52   #  ----> SHA1 password, find this with $oneuser show oneadmin
  ```

- test the EC2 interface with:

  ```
  $ euca-describe-images
  ```

[1] http://www.eucalyptus.com/download/euca2ools

There is more information in the official documentation and mailing list:

http://opennebula.org/documentation:rel4.2:ec2qcg
http://opennebula.org/documentation:rel4.2:ec2qec
http://lists.opennebula.org/pipermail/users-opennebula.org/2013-April/022550.html

X.509 authentication

Until now we have used no secure connection. OpenNebula rely on Sinatra to provide web services and this one doesn't support SSL. So we have to configure Apache as an intermediate layer (SSL proxy):

```
# yum install mod_ssl httpd httpd-tools
```

Then edit the file */etc/httpd/conf.d/ssl.conf* and add the following lines at the bottom:

```
# First part is for Sunstone
Listen 8443
SSLSessionCacheTimeout  300
SSLSessionCache         shm:/var/cache/mod_ssl/shm_cache

<VirtualHost *:8443>

SSLEngine               on
SSLCertificateFile      /etc/grid-security/hostcert.pem
SSLCertificateKeyFile   /etc/grid-security/hostkey.pem
SSLCACertificatePath    /etc/grid-security/certificates
SSLVerifyClient         optional
SSLVerifyDepth          10
SSLOptions              +ExportCertData +StdEnvVars
RequestHeader set SSL_CLIENT_CERT %{SSL_CLIENT_CERT}e
RequestHeader set SSL_CLIENT_CERT_CHAIN_0 %{SSL_CLIENT_CERT_CHAIN_0}e
RequestHeader set SSL_CLIENT_CERT_CHAIN_1 %{SSL_CLIENT_CERT_CHAIN_1}e
RequestHeader set SSL_CLIENT_CERT_CHAIN_2 %{SSL_CLIENT_CERT_CHAIN_2}e

ProxyPreserveHost off
ProxyPass        / http://localhost:9869/
ProxyPassReverse / http://localhost:9869/
</VirtualHost>


# Second part is for econe-server
Listen 8444
SSLSessionCacheTimeout  300
SSLSessionCache         shm:/var/cache/mod_ssl/shm_cache

<VirtualHost *:8444>

SSLEngine               on
SSLCertificateFile      /etc/grid-security/hostcert.pem
SSLCertificateKeyFile   /etc/grid-security/hostkey.pem
SSLCACertificatePath    /etc/grid-security/certificates
SSLVerifyClient         require
SSLVerifyDepth          10
SSLOptions              +ExportCertData +StdEnvVars
RequestHeader set SSL_CLIENT_CERT %{SSL_CLIENT_CERT}e
RequestHeader set SSL_CLIENT_CERT_CHAIN_0 %{SSL_CLIENT_CERT_CHAIN_0}e
RequestHeader set SSL_CLIENT_CERT_CHAIN_1 %{SSL_CLIENT_CERT_CHAIN_1}e
RequestHeader set SSL_CLIENT_CERT_CHAIN_2 %{SSL_CLIENT_CERT_CHAIN_2}e

ProxyPreserveHost off
ProxyPass        / http://localhost:4567/
```

```
ProxyPassReverse / http://localhost:4567/

</VirtualHost>
```

Then modify the file */etc/one/econe.conf* to use SSL connections. Server configuration and authentication section are the only important for you:

```
#############################################################
# Server Configuration
#############################################################

# Directory to store temp files when uploading images
:tmpdir: /var/tmp

# OpenNebula sever contact information
:one_xmlrpc: http://localhost:2633/RPC2

# Host and port where econe server will run
:host: localhost
:port: 4567

# SSL proxy URL that serves the API (set if is being used)
:ssl_server: https://fermicloudpp003.fnal.gov:8444/

#############################################################
# Auth
#############################################################

# Authentication driver for incomming requests
#   - ec2, default Acess key and Secret key scheme
#   - x509, for x509 certificates based authentication
:auth: x509

# Authentication driver to communicate with OpenNebula core
#   - cipher, for symmetric cipher encryption of tokens
#   - x509, for x509 certificate encryption of tokens
:core_auth: cipher
```

Then be sure to have all the files you specified in the document:

```
# ls -l /etc/grid-security/
```

The output should look similar to this (symlinks must point to existent files!):

```
total 60
drwxr-xr-x 2 root root 49152 Sep 25 11:26 certificates
-rw-r--r-- 1 root root  2045 Sep 25 10:22 fermicloudpp003.fnal.gov-hostcert.pem.new
-rw------- 1 root root  1679 Sep 25 10:22 fermicloudpp003.fnal.gov-hostkey.pem.new
lrwxrwxrwx 1 root root    57 Sep 25 11:23 hostcert.pem ->
       /etc/cloud-security/fermicloudpp003.fnal.gov-hostcert.pem
lrwxrwxrwx 1 root root    56 Sep 25 11:23 hostkey.pem ->
       /etc/cloud-security/fermicloudpp003.fnal.gov-hostkey.pem
drwxr-xr-x 2 root root  4096 Jun  3 09:28 http
```

Create a new user who uses X.509. For example, as oneadmin:

```
$ oneuser create hyunwoo "/DC=com/DC=DigiCert-Grid/O=Open Science Grid/OU=People/CN=Milan
Jovetic 765" --driver x509
```

Now you can start (or restart) your services:

```
# service httpd start
# econe-server start
```

The interface was tested with the curl command:

```
$ curl --cert certificate.pem --key keys.pem https://fermicloudpp003.fnal.gov:8444/?
Action=DescribeImages
```

The output was an empty image set:

```
<DescribeImagesResponse xmlns="http://ec2.amazonaws.com/doc//">
        <requestId>
                73db80e8-ab95-420e-8d7a-3a3bf6a9ac9d
        </requestId>
        <imagesSet>
        </imagesSet>
</DescribeImagesResponse>
```

OpenStack Grizzly EC2 interface

The RDO OpenStack installation comes with the EC2 interface enabled by default. This EC2 interface to OpenStack allows an end user to create and manage virtual machines in OpenStack using the EC2 API.

In order to use the OpenStack EC2 interface each user can source its own custom script provided by the OpenStack dashboard with all the necessary environment variables already set. Then one can use the Euca2ools binaries to make the API calls, for example *euca-describe-images*.

X.509 authentication

In this moment the X.509 authentication seems not possible, due to the fact that OpenStack doesn't support SOAP request. Fortunately the community seems aware of this problem, according to:

https://wiki.openstack.org/wiki/NOVA/EC2SoapAPISpec

https://code.launchpad.net/~soren/nova/ec2-soap-api

CloudBursting - How to burst from ONE4.2 to OpenStack Grizzly, using the EC2 interface[2]

It is possible to burst instances from OpenNebula to an EC2 compliant cloud, this allows to run instances on a public cloud when there are no resources left on FermiCloud.

Besides the default behavior of bursting to the public Amazon Web Services (AWS) cloud, it's possible to use OpenNebula to cloudburst to other clouds using their EC2 interface, here we will cover how to

[2] This procedure is based on http://opennebula.org/documentation:rel4.2:ec2g.

cloudburst to OpenStack Grizzly.

First you have to install Euca2ools on the same host where you have installed the OpenNebula front-end:

```
# yum -y install epel-release
# yum -y install --enablerepo=epel euca2ools
# #Below we make symlinks from ec2 named files to the euca* binaries. By default ONE looks for
ec2 binaries.
# mkdir -p /opt/ec2-to-euca/bin; for i in `ls /usr/bin/euca-*`; do ln -s $i
/opt/ec2-to-euca/bin/`basename $i | sed 's/euca/ec2/g'`; done[3]
# chown -R oneadmin:oneadmin /opt/ec2-to-euca/
```

Then you have to configure the OpenNebula front-end:

- edit */etc/one/oned.conf* and uncomment the following lines:

```
IM_MAD = [
       name       = "ec2",
       executable = "one_im_ec2",
       arguments  = "im_ec2/im_ec2.conf" ]

VM_MAD = [
       name       = "ec2",
       executable = "one_vmm_ec2",
       arguments  = "vmm_ec2/vmm_ec2.conf",
       type       = "xml" ]
```

- edit */etc/one/im_ec2/im_ec2.conf* to define the maximum capacity that you want to allocate in EC2:

```
#-------------------------------------------------------------------------------
# Max number of instances that can be launched into EC2
#-------------------------------------------------------------------------------

SMALL_INSTANCES=5
LARGE_INSTANCES=
EXTRALARGE_INSTANCES=
```

- edit */etc/one/vmm_ec2/vmm_ec2.conf* and make sure to have at least an m1.small template:

```
<!--
       Default configuration attributes for the EC2 driver (all domains will use these
       values as defaults)
       Valid atributes are:
              - ec2[keypair,authorizedports,instancetype]
       Use XML syntax to specify defaults, note elements are UPCASE
       Example:
              <TEMPLATE>
                     <EC2>
                            <KEYPAIR>gsg-keypair</KEYPAIR>
                            <AUTHORIZEDPORTS>22</AUTHORIZEDPORTS>
                            <INSTANCETYPE>m1.small</INSTANCETYPE>
                     </EC2>
               </TEMPLATE>
-->

<TEMPLATE>
       <EC2>
```

[3] OpenStack doesn't support EC2 official tools. So you have to pretend to use those tools with a little workaround.

```
                <INSTANCETYPE>m1.small</INSTANCETYPE>
                </EC2>
        </TEMPLATE>
```

- edit */etc/one/vmm_ec2/vmm_ec2rc* and set up all the environment variables you need:

```
#---------------------------------------------------------------------------
# EC2 API TOOLS Configuration.
#---------------------------------------------------------------------------

EC2_ACCESS_KEY=4899189e202746279ff9956181e8be79
EC2_SECRET_KEY=a92a184cdba443c2bcb1f1f7f59c7e39
EC2_URL=http://131.225.64.184:8773/services/Cloud
EC2_HOME="/opt/ec2-to-euca"

#---------------------------------------------------------------------------
# Driver configuration
#---------------------------------------------------------------------------

# Arguments for the JAVA Virtual Machine
EC2_JVM_ARGS="-Xms16m -Xmx64m"

# Number of concurrent EC2 operations (not instances)
EC2_JVM_CONCURRENCY=10
```

  You will find your authentication variables using the OpenStack GUI (Access & Security → API Access → Download EC2 Credentials), while the EC2_HOME variable is the above mentioned path to Euca2ools links.

After that you have to restart OpenNebula (as root) to apply the changes:

```
# service opennebula restart
```

Then create the EC2 host in OpenNebula. To do that create a file *system.ds* like this:

```
NAME    = ec2_ds
TM_MAD  = dummy
TYPE    = SYSTEM_DS
```

Now run the following commands:

```
# su oneadmin
$ onedatastore create system.ds
$ onecluster create ec2
$ onecluster adddatastore ec2 ec2_ds
$ onehost create ec2 --im ec2 --vm ec2 --net dummy --cluster ec2
```

Now suppose to have an image already registered in OpenStack (e.g. ami-id = ami-00000007). Create a file *vm.template* like[4]:

```
CPU="1"
EC2=[
        AMI="ami-00000007" ]
```

[4] VM templates with a EC2 section cannot define local disks (DISK=...), since this will trigger the TM drivers even if the VM is placed in a EC2 host. In other words, this time we don't need to transfer the image to the hypervisor host, because the image is supposed to be already on it.

```
MEMORY="2048"
VCPU="1"
SCHED_REQUIREMENTS = "NAME = \"ec2\""
```

And as usual (as oneadmin):

```
$ onetemplate create vm.template
$ onetemplate instantiate 0
```

After few seconds you will find the machine in the RUNNING state in both OpenStack and OpenNebula.

Unfortunately this interface is still buggy, for example:

- run the following command:

  ```
  $ onevm delete <VM ID>
  ```

  The machine will go in the DONE state as you expect. But if you check the machine log (e.g. /var/log/one/<VM ID>.log) you will see:

  ```
  Tue Sep 24 17:54:07 2013 [LCM][I]: New VM state is CLEANUP.
  Tue Sep 24 17:54:07 2013 [VMM][I]: Driver command for 96 cancelled
  Tue Sep 24 17:54:07 2013 [DiM][I]: New VM state is DONE
  Tue Sep 24 17:54:07 2013 [VMM][W]: Ignored: CLEANUP FAILURE 96 Action not implemented by
  driver EC2Driver
  ```

  As a result the VM is still in OpenStack;
- run the following command:

  ```
  $ onevm shutdown <VM ID>
  ```

  The machine will be deleted from OpenStack, but it will go in the FAILED status in OpenNebula. The log reports:

  ```
  Wed Sep 25 09:39:59 2013 [LCM][I]: New VM state is SHUTDOWN
  Wed Sep 25 09:40:00 2013 [LCM][I]: New VM state is EPILOG
  Wed Sep 25 09:40:00 2013 [TM][I]: Command execution fail:
  /var/lib/one/remotes/tm/shared/delete ec2:/var/lib/one//datastores/0/99 99 0
  Wed Sep 25 09:40:00 2013 [TM][I]: delete: Deleting /var/lib/one/datastores/0/99
  Wed Sep 25 09:40:00 2013 [TM][E]: delete: Command "rm -rf /var/lib/one/datastores/0/99"
  failed: ssh: Could not resolve hostname ec2: Name or service not known
  Wed Sep 25 09:40:00 2013 [TM][E]: Error deleting /var/lib/one/datastores/0/99
  Wed Sep 25 09:40:00 2013 [TM][I]: ExitCode: 255
  Wed Sep 25 09:40:00 2013 [TM][E]: Error executing image transfer script: Error
  deleting /var/lib/one/datastores/0/99
  Wed Sep 25 09:40:00 2013 [DiM][I]: New VM state is FAILED
  ```

OpenNebula 4.2 scheduling policies

The VM scheduling is managed using two attributes in the VM template (see http://opennebula.org/documentation:rel4.2:template#placement_section):

- SCHED_REQUIREMENTS: contains the boolean expression that will be evaluated on every host, to understand if a host is suitable or not to execute the VM;

- SCHED_RANK: contains the arithmetic expression that will be used to compute the host ranking, among suitable hosts. The host with the highest rank is chosen to execute the VM.

Remember that:

- when you don't specify the SCHED_REQUIREMENTS option, all hosts are considered suitable;
- when you don't specify the SCHED_RANK option, the default policy is contained in */etc/one/sched.conf*. In this file you can chose a standard policy (Packing, Striping, Load-aware) or a customized one;
- when two hosts have the same rank, the host chosen is the one with the highest ID.

By the way it is necessary to highlight that we didn't manage to write an ONE4.2 generic template, suitable for both EC2 and local scheduling. This because the presence of the DISK section triggers the transfer manager also if the VM is scheduled on the EC2 host (see http://opennebula.org/documentation:rel4.2:ec2g#considerations_limitations). Consequences of this are that:

- in order to cloudburst we need a template for EC2 that will never work under a normal ONE host, so the user needs to know if the VM will go or not to EC2, before instantiating it;
- we can not automate cloudburst with ONE scheduling policies.

It is important to remark also that, from previous tests, we know this limitation is not present in ONE3.2 and, since it is a known bug, it will be fixed in the next release (see http://lists.opennebula.org/pipermail/users-opennebula.org/2013-September/024678.html).

Client/server compatibility table – Access and secret keys authentication

	Amazon EC2	OpenStack Grizzly	OpenNebula 4.2

Amazon EC2 tools	🟩 Working:
	# If your Java executable is in /usr/bin, set JAVA_HOME to /usr
	JAVA_HOME=/usr
	# Path to Amazon tools
	EC2_HOME=/cloud/login/foo/ec2-api-tools-1.6.9.0/
	# Amazon endpoint, see:
	# http://docs.aws.amazon.com/general/latest/gr/rande.html#ec2_region
	EC2_URL=http://ec2.us-west-2.amazonaws.com
	# Amazon authentication credentials, see:
	# https://portal.aws.amazon.com/gp/aws/securityCredentials
	AWS_ACCESS_KEY=ACREDFXXSCFERMIZCLAB
	AWS_SECRET_KEY=1e34a67z9abhdef3vKpFAeVwurXbdeTa81ltIcDa
		🟥 Not working:
		Unexpected error: org.codehaus.xfire.fault.XFireFault: General security error; nested exception is: java.security.cert.CertificateParsingException: invalid DER-encoded certificate data
		...
			🟩 Working:
			# If your Java executable is in /usr/bin, set
			# JAVA_HOME to /usr
			JAVA_HOME=/usr
			# Path to Amazon tools
			EC2_HOME=/cloud/login/foo/ec2-api-tools-1.6.9.0/
			# ONE frontend URL, see /etc/one/econe.conf on the
			# server you want to connect to
			EC2_URL=http://fermicloudpp003.fnal.gov:4567
			# ONE username, see $ oneuser list and pick up your
			# username
			AWS_ACCESS_KEY=user
			# user SHA1 password, see $ oneuser show user
			AWS_SECRET_KEY=0beec7b5ea3f0fdbc95d0dd47f3c5bc275da8a33

Euca2ools	🟩 Working:
	# Amazon endpoint, see:
	# http://docs.aws.amazon.com/general/latest/gr/rande.html#ec2_region
	EC2_URL=http://ec2.us-west-2.amazonaws.com
	# Amazon authentication credentials, see:
	# https://portal.aws.amazon.com/gp/aws/securityCredentials
	EC2_ACCESS_KEY=ACREDFXXSCFERMIZCLAB
	EC2_SECRET_KEY=1e34a67z9abhdef3vKpFAeVwurXbdeTa81ltIcDa
		🟩 Working:
		# Just run the credentials script
		# downloaded from the dashboard
			🟩 Working:
			# ONE frontend URL, see /etc/one/econe.conf on the
			# server you want to connect to
			EC2_URL=http://fermicloudpp003.fnal.gov:4567
			# ONE username, see $ oneuser list and pick up your
			# username
			EC2_ACCESS_KEY=user
			# user SHA1 password, see $ oneuser show user
			EC2_SECRET_KEY=0beec7b5ea3f0fdbc95d0dd47f3c5bc275da8a33

ECONE tools	🟩 Working:
	# Amazon endpoint, see:
	# http://docs.aws.amazon.com/general/latest/gr/rande.html#ec2_region
	EC2_URL=http://ec2.us-west-2.amazonaws.com
	# Amazon authentication credentials, see:
	# https://portal.aws.amazon.com/gp/aws/securityCredentials
	EC2_ACCESS_KEY=ACREDFXXSCFERMIZCLAB
	EC2_SECRET_KEY=1e34a67z9abhdef3vKpFAeVwurXbdeTa81ltIcDa
		🟥 Not working:
		econe-describe-images: Not Authorized
			🟩 Working:
			# ONE frontend URL, see /etc/one/econe.conf on the
			# server you want to connect to
			EC2_URL=http://fermicloudpp003.fnal.gov:4567
			# ONE username, see $ oneuser list and pick up your
			# username
			EC2_ACCESS_KEY=user
			# user SHA1 password, see $ oneuser show user
			EC2_SECRET_KEY=0beec7b5ea3f0fdbc95d0dd47f3c5bc275da8a33

OpenStack incompatibility is due to the use of REST API instead of SOAP.

Client/server compatibility table – X.509 authentication

	Amazon EC2	OpenStack Grizzly	OpenNebula 4.2

Amazon EC2 tools	Working:
	# If your Java executable is in /usr/bin, set JAVA_HOME to /usr
	JAVA_HOME=/usr
	# Path to Amazon tools
	EC2_HOME=/cloud/login/foo/ec2-api-tools-1.6.9.0/
	# Amazon endpoint, see:
	# http://docs.aws.amazon.com/general/latest/gr/rande.html#ec2_region
	EC2_URL=http://ec2.us-west-2.amazonaws.com
	# Amazon certificate and private key, download both of them from:
	# https://portal.aws.amazon.com/gp/aws/securityCredentials
	EC2_CERT=/cloud/login/user/cert.pem
	EC2_PRIVATE_KEY=/cloud/login/user/pk.pem
		Not working:
		Unexpected error: org.codehaus.xfire.fault.XFireFault: General security error; nested exception is: java.security.cert.CertificateParsingException: invalid DER-encoded certificate data
		...
			Not working

Euca2ools	Not working: EC2_ACCESS_KEY environment variable must be set.	Not working: EC2_ACCESS_KEY environment variable must be set.	Not working (no response)
ECONE tools	Not working: econe-describe-images: AWS was not able to validate the provided access credentials	Not working: econe-describe-images: Not Authorized	Not working

Also here the spread incompatibility is due to SOAP/REST issues.